

## **REMARKS**

Receipt of the Office Action of November 6, 2008 is gratefully acknowledged.

In paragraph 3 of the Office Action, the examiner states: "In response to Applicant's argument, the Examiner submits that Clift discloses 'pointer RF entry' (Clift: Figure 4). Therefore, Clift discloses 'stack'." Clift's system, however, seems to come under the category of a register machine not a stack machine, because each entry of the RAT primary/shadow array 110/112 is to correspond to a predetermined logical register (Clift: Fig. 2, Fig. 4, column 5 lines 11-25 and column 11 lines 16-29). Clift does not disclose any stack management system. In fact, the word "stack" is not to be found in Clift's specification.

The computer system disclosed in the present application can operate in either of the two modes: the stack mode and the register mode. Each entry of the advanced mapping file (AMF 3a) is to correspond to a predetermined logical register number in the register mode, but not in the stack mode (page 33 lines 3-8).

In the stack mode of the system disclosed in the present application, the entry of address 0 of AMF 3a always corresponds to the top of the stack, and the entry of address n of AMF 3a corresponds to the (n+1)th element of the stack (page 14 line 23 – page 16 line 13 and Fig. 3). On the other hand, traditional look-ahead stack management systems employ the circular buffer technique, which requires two pointers to indicate the top and the bottom.

In accordance with the difference of the structure described above, AMF 3a is to be manipulated in a peculiar manner in the stack mode of the system disclosed in the present application (page 23 line 21 – page 24 line 21).

On page 28 line 1 – page 32 line 25 of the specification, an example action of processing two instructions, Instruction\_1 and Instruction\_2, is described. As described on page 8 line 11 – page 11 line 7, ten stack instructions have been converted into Instruction\_1 and Instruction\_2. The state-modification field content of Instruction\_1, SM{ +2: f4, f1 }, implies that the stack is to be grown by two elements, which are to correspond to f4 and f1 (f1 - f4 represent addresses of the data-file entries to be allocated to hold result data). And, the state-modification field content of Instruction\_2, SM{ -2: }, implies that the stack is to be shrunk by two elements.

In the example action, when the computer system is in such a state as shown in Fig. 8, Instruction\_1 is issued and the AMF 3a is so modified that p26 and p51, replacing f1 and f4, are respectively entered into the entries of address 0 and 1, and for the part below, contents of the AMF entries are shifted down by the amount of stack growth (2 entries). (The contents of the AMF entries of address 0, 1, ... shown in Fig. 8 are moved into the AMF entries of address 2, 3, ..., respectively.) The state of the computer system right after issue of Instruction\_1 is shown in Fig. 9. The equivalent action of processing Instruction\_1 in a traditional system is illustrated in Application No.10/344,698: page 38 lines 13-23, Fig. 13 and Fig. 14.

In the next cycle, Instruction\_2 is issued. As only a negative growth of the stack (-2) is indicated in the SM field, contents of the AMF entries are shifted by this amount. (The contents of the AMF entries of address 2, 3, ... shown in Fig. 9 are moved into the AMF entries of address 0, 1, ..., respectively.) The state of the computer system right after issue of Instruction\_2 is shown in Fig. 10. The equivalent action of processing Instruction\_2 in a traditional system is illustrated in Application No.10/344,698: page 39 line 15 – page 40 line 1, Fig. 14 and Fig. 15.

Claims 6-9 have been examined with the following result: (1) claims 6-9 are finally rejected under 35 USC 112, second paragraph as indefinite; and (2) claims 6-9 are finally rejected under 35 USC 102(b) by Clift et al.

(1)

Regarding (1), claims 6-9 have been amended to clarify intent without changing the scope of the claims. The present amendments should be entered as they are believed to clarify the claims on the points raised by the examiner.

Here, it should be noted that the data file (DF 6) and the advanced mapping file (AMF 3a) in the specification are respectively referred to as "data storing means" and "look-ahead mapping means" in claims 6-9.

As per claim 6, the phrase "for each entry of said look-ahead mapping means that is to hold an entry address in said data storing means allocated to an operand stack element" in lines 9-11 has been inserted in order to exclude entries that is to be below the bottom (shaded area of AMF 3a in Fig. 3), and "the entry" in lines 11-12 refers to "each entry" in line 9.

As per claim 7, the phrase "for each entry of said look-ahead mapping means holding an entry address in said data storing means allocated to an operand stack element" in lines 9-11 has been inserted in order to exclude entries that is below the bottom right before the modification. The phrase "if the entry of said look-ahead mapping means is to hold an entry address in said data storing means allocated to an operand stack element" in lines 11-13 has been inserted in order to exclude cases where "the entry", which refers to "each entry" in line 9, is to fall below the bottom right after the modification. The phrase "whose value is held / to be held in the entry of said data storing means indicated by the address held in the entry of said look-ahead mapping means" in lines 14-16 qualifies "the operand stack

element" in line 14; and the phrase "indicated by the address held in the entry of said look-ahead mapping means" in lines 15-16 qualifies "the entry of said data storing means" in line 15. And, "the entry" in line 16 also refers to "each entry" in line 9. Hence, "the number of operand stack elements over the operand stack element whose ..." is to be unchanged. Since "look-ahead mapping means" indicates the look-ahead state of the system, the entry of the data storing means indicated by an address held in the look-ahead mapping means may or may not hold the value at the time of the modification. So, the expression of "held / to be held" in lines 14-15 can be considered to be appropriate.

Claims 8 and 9 are respectively identical to claims 6 and 7 except that "entry"/"entries" is changed to "register"/"registers", and accordingly, "(entry) address" is changed to "(register) number".

The above explanation and the amendments should obviate the rejection under 35 USC 112.

(2)

Regarding (2), it is noted that Clift et al never discloses any look-ahead stack management system, and cannot, it is respectfully submitted therefore, anticipate claims 6-9.

The circuit for making a modification on a look-ahead mapping means can be streamlined by adopting the look-ahead stack management system according to claims 6-9. It is not possible to achieve this result with a traditional stack management system equipped with a circular buffer. And, the look-ahead mapping means can have a non-power-of-two number of entries (registers) for stack management.

In view of the foregoing, reconsideration and re-examination are respectfully requested and the noted amendments to the claims entered and claims 6-9 found allowable. Alternatively, entry of the amendments to claims 6-9 is respectfully requested for purposes of appeal.

If there is an additional fee (for claims or extensions of time) required for this paper, please charge the fee to our **Deposit Account Number 02-0200**.

Respectfully submitted,  
BACON & THOMAS, PLLC



Felix J. D'Ambrosio  
Registration No: 25,721

Date: March 6, 2009

**Customer Number \*23364\***

BACON & THOMAS  
625 Slaters Lane, Fourth Floor  
Alexandria, Virginia 22314  
Phone: (703) 683-0500

C:\Documents and Settings\YAMOTO\Local Settings\Temporary Internet Files\OLK15\March 4 2009 Response.wpd